

[illegible]

FFFFFFFFF	000000	RRRRRRRR	DDDDDDDD	EEEEEEEEEE	LL	EEEEEEEEEE	TTTTTTTTTT	EEEEEEEEEE	
FFFFFFFFF	000000	RRRRRRRR	DDDDDDDD	EEEEEEEEEE	LL	EEEEEEEEEE	TTTTTTTTTT	EEEEEEEEEE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FFFFFFFFF	00	RRRRRRRR	DD	EEEEEEEEEE	LL	EEEEEEEEEE	TT	EEEEEEEEEE	
FFFFFFFFF	00	RRRRRRRR	DD	EEEEEEEEEE	LL	EEEEEEEEEE	TT	EEEEEEEEEE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	00	RR	DD	EE	LL	EE	TT	EE	
FF	000000	RR	DDDDDDDD	EEEEEEEEEE	LLLLLLLLLL	EEEEEEEEEE	TT	EEEEEEEEEE
FF	000000	RR	DDDDDDDD	EEEEEEEEEE	LLLLLLLLLL	EEEEEEEEEE	TT	EEEEEEEEEE

LL	IIIIII	SSSSSSSS	
LL	IIIIII	SSSSSSSS	
LL	II	SS	
LL	II	SS	
LL	II	SS	
LL	II	SS	
LL	II	SSSSSS	
LL	II	SSSSSS	
LL	II	SS	SS
LL	II	SS	SS
LL	II	SS	SS
LL	II	SS	SS
LLLLLLLLLL	IIIIII	SSSSSSSS	
LLLLLLLLLL	IIIIII	SSSSSSSS	

```
1 0001 0 MODULE FOR$DELETE ( ! DELETE statement processor
2 0002 0 IDENT = '1-002' ! Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: FORTRAN Language Support Library
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 Contains routines to implement FORTRAN DELETE for relative
37 0037 1 and indexed organization files.
38 0038 1
39 0039 1 ENVIRONMENT: User mode, AST reentrant
40 0040 1
41 0041 1 AUTHOR: Steven B. Lionel, CREATION DATE: 14-May-1979
42 0042 1
43 0043 1 EDIT HISTORY:
44 0044 1
45 0045 1 1-001 - Original. SBL 14-May-1979
46 0046 1 1-002 - Move ACTUALCOUNT declaration inside routine. SBL 15-June-1982
47 0047 1 --
```



```

49      0048 1  |
50      0049 1  | SWITCHES:
51      0050 1  |
52      0051 1  |
53      0052 1  | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
54      0053 1  |
55      0054 1  |
56      0055 1  | LINKAGES:
57      0056 1  |
58      0057 1  |
59      0058 1  | REQUIRE 'RTLIN:OTSLNK';           ! Define all linkages
60      0487 1  |
61      0488 1  |
62      0489 1  | TABLE OF CONTENTS:
63      0490 1  |
64      0491 1  |
65      0492 1  | FORWARD ROUTINE
66      0493 1  |     FOR$DELETE : CALL CCB,           ! Sequential access delete entry
67      0494 1  |     FOR$DELETE_D : CALL_CCB;        ! Direct access delete entry
68      0495 1  |
69      0496 1  |
70      0497 1  | INCLUDE FILES:
71      0498 1  |
72      0499 1  |
73      0500 1  | LIBRARY 'RTLSTARLE';             ! STARLET library for macros and symbols
74      0501 1  | REQUIRE 'RTLML:FORERR';          ! FORTRAN error numbers
75      0569 1  | REQUIRE 'RTLML:OTSLUB';          ! Logical Unit Block Definitions
76      0709 1  | REQUIRE 'RTLML:OTSISB';          ! Inter-statement Block Definitions
77      0877 1  | REQUIRE 'RTLIN:RTLPSECT';        ! P-SECT declaration macros
78      0972 1  | REQUIRE 'RTLML:FORPAR';          ! Inter-module parameters
79      0995 1  |
80      0996 1  |
81      0997 1  | MACROS:
82      0998 1  |
83      0999 1  |
84      1000 1  |
85      1001 1  | EQUATED SYMBOLS:
86      1002 1  |
87      1003 1  |
88      1004 1  |
89      1005 1  | PSECT DEFINITIONS:
90      1006 1  |
91      1007 1  |
92      1008 1  | DECLARE_PSECTS (FOR);
93      1009 1  |
94      1010 1  |
95      1011 1  | OWN STORAGE:
96      1012 1  |
97      1013 1  |
98      1014 1  |
99      1015 1  | EXTERNAL REFERENCES:
100     1016 1  |
101     1017 1  |
102     1018 1  | EXTERNAL ROUTINE
103     1019 1  |     FOR$$IOSTAT_HND,           ! IOSTAT error handler
104     1020 1  |     FOR$$SIGNAL_STO : NOVALUE, ! Signal_stop routine
105     1021 1  |     FOR$$SIG_NO_LUB : NOVALUE, ! Signal without LUB
```

FOR\$DELETE
1-002

M 15
16-Sep-1984 00:17:29
14-Sep-1984 12:31:48

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORDELETE.B32;1

Page 3
(2)

: 106
: 107

1022 1
1023 1

FOR\$\$CB_PUSH : JSB_CB_PUSH NOVALUE, ! Push a CCB
FOR\$\$CB_POP : JSB_CB_POP NOVALUE; ! Pop a CCB

```
109 1024 1 GLOBAL ROUTINE FOR$DELETE (
110 1025 1     UNIT,                                ! Unit number on which to delete
111 1026 1     ERR_EQL                          ! 1 if ERR= or IOSTAT= present
112 1027 1 ) : CALL_CCB =
113 1028 1
114 1029 1 ++
115 1030 1 FUNCTIONAL DESCRIPTION:
116 1031 1
117 1032 1 FOR$DELETE deletes the current record of the indexed or relative organization file
118 1033 1 open on unit UNIT.
119 1034 1
120 1035 1 CALLING SEQUENCE:
121 1036 1
122 1037 1     iostat.wl.v = FOR$DELETE (unit.rl.v [, err_eql.rlu.v])
123 1038 1
124 1039 1 FORMAL PARAMETERS:
125 1040 1
126 1041 1     unit                                - The unit number on which to delete the record.
127 1042 1                                     There must be a "current record" on this unit.
128 1043 1     err_eql                            - If absent or zero, all errors are signalled.
129 1044 1                                     - If 1, errors return an IOSTAT error code value.
130 1045 1
131 1046 1 IMPLICIT INPUTS:
132 1047 1
133 1048 1     CCB
134 1049 1
135 1050 1 IMPLICIT OUTPUTS:
136 1051 1
137 1052 1     NONE
138 1053 1
139 1054 1 ROUTINE VALUE:
140 1055 1
141 1056 1     An IOSTAT value.
142 1057 1
143 1058 1 SIDE EFFECTS:
144 1059 1
145 1060 1     SIGNAL STOP's:
146 1061 1     FOR$WRITEAFIL - Write to read-only file
147 1062 1     FOR$NO CURREC - No current record
148 1063 1     FOR$DE[ERR   - Delete error
149 1064 1
150 1065 1 --
151 1066 1
152 1067 2 BEGIN
153 1068 2
154 1069 2 GLOBAL REGISTER
155 1070 2     CCB = 11: REF BLOCK [, BYTE];
156 1071 2
157 1072 2 LOCAL
158 1073 2     L_UNWIND_ACTION : VOLATILE,
159 1074 2     L_ERR_EQ[_PRES : VOLATILE,
160 1075 2     STATUS;
161 1076 2
162 1077 2 BUILTIN
163 1078 2     ACTUALCOUNT;
164 1079 2
165 1080 2 ENABLE
```



```
166 1081 FOR$$IOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);
167 1082
168 1083 !+
169 1084 ! Determine if ERR_EQL is present.
170 1085 !-
171 1086
172 1087 IF ACTUALCOUNT () GTR 1
173 1088 THEN
174 1089     L_ERR_EQL_PRES = .ERR_EQL
175 1090 ELSE
176 1091     L_ERR_EQL_PRES = 0;
177 1092
178 1093 !+
179 1094 ! Unwind action is NO-OP (no LUB yet)
180 1095 !-
181 1096
182 1097 L_UNWIND_ACTION= FOR$K_UNWINDNOP;
183 1098
184 1099 !+
185 1100 ! Get a LUB for this unit. On return, CCB points to the
186 1101 ! current control block.
187 1102 !-
188 1103
189 1104 FOR$$CB_PUSH (.UNIT, LUB$K_LUN_MIN);
190 1105
191 1106 !+
192 1107 ! Unwind action is now to POP the CCB.
193 1108 !-
194 1109
195 1110 L_UNWIND_ACTION = FOR$K_UNWINDPOP;
196 1111
197 1112 !+
198 1113 ! If file is not indexed or relative organization, or is direct access,
199 1114 ! signal error FOR$DELEERR.
200 1115 !-
201 1116
202 1117 IF NOT .CCB [LUB$V_NOTSEQORG] OR .CCB [LUB$V_DIRECT]
203 1118 THEN
204 1119     FOR$$SIGNAL_STO (FOR$K_DELEERR);
205 1120
206 1121 !+
207 1122 ! If file is read-only, signal error FOR$WRIREFIL.
208 1123 !-
209 1124
210 1125 IF .CCB [LUB$V_READ_ONLY]
211 1126 THEN
212 1127     FOR$$SIGNAL_STO (FOR$K_WRIREFIL);
213 1128
214 1129 !+
215 1130 ! Try to delete the current record. If we get an error, signal it.
216 1131 !-
217 1132
218 1133 IF NOT $DELETE (RAB=.CCB)
219 1134 THEN
220 1135     BEGIN
221 1136     WHILE .CCB [RAB$L_STS] EQL RMSS_RSA DO
222 1137     BEGIN
```

```

223      1138      4      $WAIT (RAB=.CCB);
224      1139      4      $DELETE (RAB=.CCB);
225      1140      4      END;
226      1141      4      IF NOT .CCB [RAB$L_STS]
227      1142      4      THEN
228      1143      4          FOR$$SIGNAL STO (
229      1144      4              SELECTONEU .CCB [RAB$L_STS] OF
230      1145      4                  SET
231      1146      4                      [RMSS$ CUR, RMSS$ RNL] : FOR$K_NO_CURREC;
232      1147      4                      [OTHERWISE] : FOR$K_DELEERR;
233      1148      4                      TES);
234      1149      4      END;
235      1150      4
236      1151      4      !+
237      1152      4      ! Return I/O system to previous state
238      1153      4      !-
239      1154      4
240      1155      4      FOR$$CB_POP ();
241      1156      4
242      1157      4      RETURN 0;          ! Success IOSTAT value
243      1158      4      END;

```

				.TITLE	FOR\$DELETE	
				.IDENT	\1-002\	
				.EXTRN	FOR\$\$IOSTAT_HND	
				.EXTRN	FOR\$\$SIGNAL_STO	
				.EXTRN	FOR\$\$SIG NO-LUB	
				.EXTRN	FOR\$\$CB_PUSH, FOR\$\$CB_POP	
				.EXTRN	SYSS\$DELETE, SYSS\$WAIT	
				.PSECT	_FOR\$CODE, NOWRT, SHR, PIC, 2	
				.ENTRY	FOR\$DELETE, Save R2, R3, R4, R11	1024
54	00000000G	00	9E 00002	MOVAB	SYSS\$DELETE, R4	
53	00000000G	00	9E 00009	MOVAB	FOR\$\$SIGNAL_STO, R3	
5E		04	C2 00010	SUBL2	#4, SP	
		7E	D4 00013	CLRL	L_ERR_EQL PRES	1067
	04	AE	D4 00015	CLRL	L_UNWIND ACTION	
6D	0088	CF	DE 00018	MOVAL	12\$, (FP)	
01		6C	91 0001D	CMPB	(AP), #1	1087
		06	1B 00020	BLEQU	1\$	
6E	08	AC	D0 00022	MOVL	ERR_EQL, L_ERR_EQL_PRES	1089
		02	11 00026	BRB	2\$	
04	AE	6E	D4 00028 1\$:	CLRL	L_ERR_EQL PRES	1091
		01	D0 0002A 2\$:	MOVL	#T, L_UNWIND_ACTION	1097
		50	D4 0002E	CLRL	R0	1104
52	04	AC	D0 00030	MOVL	UNIT, R2	
	00000000G	00	16 00034	JSB	FOR\$\$CB_PUSH	
	04	AE	D4 0003A	CLRL	L_UNWIND ACTION	1110
05	A1	AB	03	BBC	#3, -95(CCB), 3\$	1117
05	FC	AB	04	BBC	#4, -4(CCB), 4\$	
		37	DD 00047 3\$:	PUSHL	#55	1119
		01	FB 00049	CALLS	#1, FOR\$\$SIGNAL_STO	
05	FC	AB	02	BBC	#2, -4(CCB), 5\$	1125
		2F	DD 00051 4\$:	PUSHL	#47	1127

FOR\$DELETE
1-002

D 16
16-Sep-1984 00:17:29
14-Sep-1984 12:31:48

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORDELETE.B32;1

Page 7
(3)

63		01	FB	00053	CALLS	#1, FOR\$\$SIGNAL_STO	:	1133
		5B	DD	00056	PUSHL	CCB	:	
64		01	FB	00058	CALLS	#1, SYSS\$DELETE	:	
3D		50	E8	0005B	BLBS	R0, 11\$:	
000182DA	8F	08	AB	D1 0005E	6\$: CMPL	8(CCB), #99034	:	1136
		10	12	00066	BNEQ	7\$:	
		5B	DD	00068	PUSHL	CCB	:	1138
00000000G	00	01	FB	0006A	CALLS	#1, SYSS\$WAIT	:	
		5B	DD	00071	PUSHL	CCB	:	1139
64		01	FB	00073	CALLS	#1, SYSS\$DELETE	:	
		E6	11	00076	BRB	6\$:	1136
1F		08	AB	E8 00078	7\$: BLBS	8(CCB), 11\$:	1141
50		08	AB	D0 0007C	MOVL	8(CCB), R0	:	
000181A0	8F		50	D1 00080	CMPL	R0, #98720	:	1144
			09	13 00087	BEQL	8\$:	1146
000184B4	8F		50	D1 00089	CMPL	R0, #99508	:	
			04	12 00090	BNEQ	9\$:	
			35	DD 00092	8\$: PUSHL	#53	:	
			02	11 00094	BRB	10\$:	
			37	DD 00096	9\$: PUSHL	#55	:	1147
63		01	FB	00098	10\$: CALLS	#1, FOR\$\$SIGNAL_STO	:	1144
	00000000G	00	16	0009B	11\$: JSB	FOR\$\$CB_POP	:	1155
		50	D4	000A1	CLRL	R0	:	1157
			04	000A3	RET		:	1158
			0000	000A4	12\$: .WORD	Save nothing	:	1067
50		08	AC	D0 000A6	MOVL	8(AP), R0	:	
50		04	A0	D0 000AA	MOVL	4(R0), R0	:	
		F8	A0	9F 000AE	PUSHAB	L_ERR EQL PRES	:	
		FC	A0	9F 000B1	PUSHAB	L_UNWIND_ACTION	:	
			02	DD 000B4	PUSHL	#2	:	
			5E	DD 000B6	PUSHL	SP	:	
7E		04	AC	7D 000B8	MOVQ	4(AP), -(SP)	:	
00000000G	00		03	FB 000BC	CALLS	#3, FOR\$\$IOSTAT_HND	:	
			04	000C3	RET		:	

; Routine Size: 196 bytes, Routine Base: _FOR\$CODE + 0000

```
245 1159 1 GLOBAL ROUTINE FOR$DELETE_D (
246 1160 1     UNIT,                ! Unit to delete on
247 1161 1     REC_NO,          ! Record number to delete
248 1162 1     ERR_EQL,         ! 1 if ERR= or IOSTAT= specified
249 1163 1     ) : CALL_CCB =
250 1164 1
251 1165 1 ++
252 1166 1 FUNCTIONAL DESCRIPTION:
253 1167 1
254 1168 1 Deletes the specified record on a relative organization file
255 1169 1 opened for direct access.
256 1170 1
257 1171 1 CALLING SEQUENCE:
258 1172 1
259 1173 1     iostat.wl.v = FOR$DELETE_D (unit.rl.v, rec_no.rl.v [, err_eql.r;u.v])
260 1174 1
261 1175 1 FORMAL PARAMETERS:
262 1176 1
263 1177 1     unit                - Unit number to delete on
264 1178 1     rec_no              - Record number to delete
265 1179 1     err_eql             - If present and 1, return IOSTAT
266 1180 1                      values for errors.
267 1181 1
268 1182 1 IMPLICIT INPUTS:
269 1183 1
270 1184 1     CCB
271 1185 1     LUB$LOG_RECNO
272 1186 1     LUB$V_FIND_LAST    ! On if FIND was last operation
273 1187 1
274 1188 1 IMPLICIT OUTPUTS:
275 1189 1
276 1190 1     LUB$V_FIND_LAST    ! Cleared
277 1191 1
278 1192 1 ROUTINE VALUE:
279 1193 1
280 1194 1     An IOSTAT value.
281 1195 1
282 1196 1 SIDE EFFECTS:
283 1197 1
284 1198 1     SIGNAL STOPS:
285 1199 1     FOR$OPEDEFREQ - Open or define file required for direct or keyed access
286 1200 1     FOR$WRIREFIL  - Write to read-only file
287 1201 1     FOR$RECNUMOUT - Record number out of range
288 1202 1     FOR$ATTACNON  - Attempt to access non-existent record
289 1203 1     FOR$SPERECLOC - Specified record locked
290 1204 1     FOR$DELERR    - Delete error
291 1205 1
292 1206 1 --
293 1207 1
294 1208 2 BEGIN
295 1209 2
296 1210 2 GLOBAL REGISTER
297 1211 2     CCB = 11: REF BLOCK [, BYTE];
298 1212 2
299 1213 2 LOCAL
300 1214 2     L_UNWIND_ACTION : VOLATILE,
301 1215 2     L_ERR_EQL_PRES : VOLATILE;
```

```

302      1216      2
303      1217      2
304      1218      2
305      1219      2
306      1220      2
307      1221      2
308      1222      2
309      1223      2
310      1224      2
311      1225      2
312      1226      2
313      1227      2
314      1228      2
315      1229      2
316      1230      2
317      1231      2
318      1232      2
319      1233      2
320      1234      2
321      1235      2
322      1236      2
323      1237      2
324      1238      2
325      1239      2
326      1240      2
327      1241      2
328      1242      2
329      1243      2
330      1244      2
331      1245      2
332      1246      2
333      1247      2
334      1248      2
335      1249      2
336      1250      2
337      1251      2
338      1252      2
339      1253      2
340      1254      2
341      1255      2
342      1256      2
343      1257      2
344      1258      2
345      1259      2
346      1260      2
347      1261      2
348      1262      2
349      1263      2
350      1264      2
351      1265      2
352      1266      2
353      1267      2
354      1268      2
355      1269      2
356      1270      2
357      1271      2
358      1272      2

BUILTIN
  ACTUALCOUNT;

ENABLE
  FOR$$IOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);

!+
! Determine if ERR_EQL is present.
!-

IF ACTUALCOUNT () GTR 2
THEN
  L_ERR_EQL_PRES = .ERR_EQL
ELSE
  L_ERR_EQL_PRES = 0;

!+
! Unwind action is NO-OP (no LUB yet)
!-

L_UNWIND_ACTION= FOR$K_UNWINDNOP;

!+
! Get a LUB for this unit. On return, CCB points to the
! current control block.
!-

FOR$$CB_PUSH (.UNIT, LUB$K_LUN_MIN);

!+
! Unwind action is now to POP the CCB.
!-

L_UNWIND_ACTION = FOR$K_UNWINDPOP;

!+
! If file is not relative organization, signal error FOR$_DELERR.
!-

IF .CCB [LUB$B_ORGAN] NEQU LUB$K_ORG_RELAT
THEN
  FOR$$SIGNAL_STO (FOR$K_DELERR);

!+
! If file is read-only, signal error FOR$_WRIREADFIL.
!-

IF .CCB [LUB$V_READ_ONLY]
THEN
  FOR$$SIGNAL_STO (FOR$K_WRIREADFIL);

!+
! If file is not direct access, signal error FOR$_OPEDEFREQ.
!-

IF NOT .CCB [LUB$V_DIRECT]
```



```
359 1273 2 THEN
360 1274 FOR$$SIGNAL_STO (FOR$K_OPEDEFREQ);
361 1275
362 1276 !+
363 1277 ! If the record number passed is illegal, signal FOR$_RECNUMOUT.
364 1278 !-
365 1279
366 1280 IF .REC_NO LEQ 0 OR (.CCB [LUB$$_LOG_REC_MAX] NEQ 0 AND .REC_NO GTRU .CCB [LUB$$_LOG_REC_MAX])
367 1281 THEN
368 1282 FOR$$SIGNAL_STO (FOR$K_RECNUMOUT);
369 1283
370 1284 !+
371 1285 ! See if the requested record to be deleted may already be locked.
372 1286 ! If it is, we want to avoid doing a $FIND which would unlock the
373 1287 ! record. To check, we compare REC_NO against (LUB$$_LOG_RECNO - 1)
374 1288 ! (or LUB$$_LOG_RECNO if LUB$$_FIND_LAST is set). If they are equal,
375 1289 ! the last operation we did may have locked this record. If it didn't,
376 1290 ! then we don't have a record locked. Trying to delete the record
377 1291 ! will show if it's locked or not. If not, we do a $FIND anyway.
378 1292 !-
379 1293
380 1294 IF .REC_NO EQL
381 1295 ( IF .CCB [LUB$$_FIND_LAST]
382 1296 THEN
383 1297 .CCB [LUB$$_LOG_RECNO]
384 1298 ELSE
385 1299 .CCB [LUB$$_LOG_RECNO] - 1)
386 1300 THEN
387 1301 BEGIN
388 1302 CCB [LUB$$_FIND_LAST] = 0; ! Turn off bit
389 1303
390 1304 !+
391 1305 ! We may have it locked. Try to delete it.
392 1306 !-
393 1307
394 1308 IF NOT $DELETE (RAB=.CCB)
395 1309 THEN
396 1310 WHILE .CCB [RAB$$_STS] EQL RMSS_RSA DO
397 1311 BEGIN
398 1312 $WAIT (RAB=.CCB);
399 1313 $DELETE (RAB=.CCB);
400 1314 END;
401 1315
402 1316 !+
403 1317 ! If we succeeded, return.
404 1318 !-
405 1319
406 1320 IF .CCB [RAB$$_STS]
407 1321 THEN
408 1322 BEGIN
409 1323 FOR$$CB_POP (); ! Return I/O to previous state
410 1324 RETURN 0; ! Success IOSTAT value
411 1325 END;
412 1326
413 1327 END;
414 1328
415 1329 !+
```

FOR\$DELETE
1-002

H 16
16-Sep-1984 00:17:29
14-Sep-1984 12:31:48

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORDELETE.B32;1

Page 11
(4)

```

416 1330 2  !- We don't have it locked. Do a $FIND then a $DELETE.
417 1331 2  !-
418 1332 2
419 1333 2  CCB [LUB$V_FIND_LAST] = 0;
420 1334 2
421 1335 2  CCB [LUB$L_LOG_RECNO] = .REC_NO;
422 1336 2  IF NOT $FIND (RAB=.CCB)
423 1337 2  THEN
424 1338 2  BEGIN
425 1339 2  WHILE .CCB[RAB$L_STS] EQL RMSS_RSA DO
426 1340 2  BEGIN
427 1341 2  $WAIT (RAB=.CCB);
428 1342 2  $FIND (RAB=.CCB)
429 1343 2  END;
430 1344 2  IF NOT .CCB [RAB$L_STS] THEN FOR$$SIGNAL_STO (
431 1345 2  SELECTONEU .CCB [RAB$L_STS] OF
432 1346 2  SET
433 1347 2  [RMSS_RLK] : FOR$K SPERECLOC;
434 1348 2  [RMSS_RNF, RMSS_EOF] : FOR$K ATTACCNON;
435 1349 2  [OTHERWISE] : FOR$K_DELERR;
436 1350 2  TES);
437 1351 2  END;
438 1352 2
439 1353 2  !+
440 1354 2  !- Try to delete the current record. If we get an error, signal it.
441 1355 2  !-
442 1356 2
443 1357 2  IF NOT $DELETE (RAB=.CCB)
444 1358 2  THEN
445 1359 2  BEGIN
446 1360 2  WHILE .CCB [RAB$L_STS] EQL RMSS_RSA DO
447 1361 2  BEGIN
448 1362 2  $WAIT (RAB=.CCB);
449 1363 2  $DELETE (RAB=.CCB);
450 1364 2  END;
451 1365 2  IF NOT .CCB [RAB$L_STS]
452 1366 2  THEN
453 1367 2  FOR$$SIGNAL_STO (FOR$K_DELERR);
454 1368 2  END;
455 1369 2
456 1370 2  !+
457 1371 2  !- Return I/O system to previous state
458 1372 2  !-
459 1373 2
460 1374 2  FOR$$CB_POP ();
461 1375 2
462 1376 2  RETURN 0; ! Success IOSTAT value
463 1377 2  END;
```

.EXTRN SYSS\$FIND

.ENTRY FOR\$DELETE D, Save R2,R3,R4,R5,R6,R11
MOVAB SYSS\$FIND, R6
MOVAB SYSS\$WAIT, R5
MOVAB SYSS\$DELETE, R4

087C 00000
56 00000000G 00 9E 00002
55 00000000G 00 9E 00009
54 00000000G 00 9E 00010

: 1159
:
:
:

		53	00000000G	00	9E	00017	MOVAB	FOR\$\$SIGNAL_STO, R3	
		5E		04	C2	0001E	SUBL2	#4, SP	1208
			04	7E	D4	00021	CLRL	L_ERR_EQL PRES	
		6D	0117	AE	D4	00023	CLRL	L_UNWIND_ACTION	
		02		CF	DE	00026	MOVAL	23\$, (FP)	1227
				6C	91	0002B	CMPB	(AP), #2	
		6E	0C	06	1B	0002E	BLEQU	1\$	1229
				AC	D0	00030	MOVL	ERR_EQL, L_ERR_EQL_PRES	
				02	11	00034	BRB	2\$	
				6E	D4	00036	CLRL	L_ERR_EQL PRES	1231
	04	AE		01	D0	00038	MOVL	#T, L_UNWIND_ACTION	1237
				50	D4	0003C	CLRL	R0	1244
	52		04	AC	D0	0003E	MOVL	UNIT, R2	
		00000000G		00	16	00042	JSB	FOR\$\$CB PUSH	
			04	AE	D4	00048	CLRL	L_UNWIND_ACTION	1250
	02		C4	AB	91	0004B	CMPB	-80(CCB), #2	1256
				05	13	0004F	BEQL	3\$	
				37	DD	00051	PUSHL	#55	1258
		63		01	FB	00053	CALLS	#1, FOR\$\$SIGNAL_STO	
05	FC	AB		02	E1	00056	BBC	#2, -4(CCB), 4\$	1264
				2F	DD	0005B	PUSHL	#47	1266
		63		01	FB	0005D	CALLS	#1, FOR\$\$SIGNAL_STO	
05	FC	AB		04	E0	00060	BBS	#4, -4(CCB), 5\$	1272
				1A	DD	00065	PUSHL	#26	1274
		63		01	FB	00067	CALLS	#1, FOR\$\$SIGNAL_STO	
		52	08	AC	D0	0006A	MOVL	REC_NO, R2	1280
				0B	15	0006E	BLEQ	6\$	
			E4	AB	D5	00070	TSTL	-28(CCB)	
				0B	13	00073	BEQL	7\$	
	E4	AB		52	D1	00075	CMPL	R2, -28(CCB)	
				05	1B	00079	BLEQU	7\$	
				19	DD	0007B	PUSHL	#25	1282
		63		01	FB	0007D	CALLS	#1, FOR\$\$SIGNAL_STO	
06	A0	AB		03	E1	00080	BBC	#3, -96(CCB), 8\$	1295
		50	E0	AB	D0	00085	MOVL	-32(CCB), R0	1297
				05	11	00089	BRB	9\$	
50	E0	AB		01	C3	0008B	SUBL3	#1, -32(CCB), R0	1299
		50		52	D1	00090	CMPL	R2, R0	1295
				26	12	00093	BNEQ	12\$	
	A0	AB		08	8A	00095	BICB2	#8, -96(CCB)	1303
				5B	DD	00099	PUSHL	CCB	1309
		64		01	FB	0009B	CALLS	#1, SYSS\$DELETE	
		16		50	E8	0009E	BLBS	R0, 11\$	
000182DA	8F		08	AB	D1	000A1	CMPL	8(CCB), #99034	1311
				0C	12	000A9	BNEQ	11\$	
				5B	DD	000AB	PUSHL	CCB	1313
		65		01	FB	000AD	CALLS	#1, SYSS\$WAIT	
				5B	DD	000B0	PUSHL	CCB	1314
		64		01	FB	000B2	CALLS	#1, SYSS\$DELETE	
				EA	11	000B5	BRB	10\$	1311
		7D	08	AB	E8	000B7	BLBS	8(CCB), 22\$	1321
	A0	AB		08	8A	000BB	BICB2	#8, -96(CCB)	1333
	E0	AB		52	D0	000BF	MOVL	R2, -32(CCB)	1335
				5B	DD	000C3	PUSHL	CCB	1336
		66		01	FB	000C5	CALLS	#1, SYSS\$FIND	
		46		50	E8	000C8	BLBS	R0, 19\$	
000182DA	8F		08	AB	D1	000CB	CMPL	8(CCB), #99034	1339

		0C	12	000D3	BNEQ	14\$		
		5B	DD	000D5	PUSHL	CCB		1341
	65	01	FB	000D7	CALLS	#1, SYSSWAIT		
		5B	DD	000DA	PUSHL	CCB		1342
	66	01	FB	000DC	CALLS	#1, SYSS\$FIND		
		EA	11	000DF	BRB	13\$		
	2C	08	AB	E8 000E1	BLBS	8(CCB), 19\$		1344
	50	08	AB	D0 000E5	MOVL	8(CCB), R0		1345
000182AA	8F		50	D1 000E9	CMPL	R0, #98986		1347
			04	12 000F0	BNEQ	15\$		
			34	DD 000F2	PUSHL	#52		
			18	11 000F4	BRB	18\$		
0001827A	8F		50	D1 000F6	CMPL	R0, #98938		1348
000182B2	8F		09	13 000FD	BEQL	16\$		
			50	D1 000FF	CMPL	R0, #98994		
			04	12 00106	BNEQ	17\$		
			24	DD 00108	PUSHL	#36		
			02	11 0010A	BRB	18\$		
			37	DD 0010C	PUSHL	#55		1349
	63		01	FB 0010E	CALLS	#1, FOR\$\$\$SIGNAL_STO		1345
			5B	DD 00111	PUSHL	CCB		1357
	64		01	FB 00113	CALLS	#1, SYSS\$DELETE		
	1F		50	E8 00116	BLBS	R0, 22\$		
000182DA	8F	08	AB	D1 00119	CMPL	8(CCB), #99034		1360
			0C	12 00121	BNEQ	21\$		
			5B	DD 00123	PUSHL	CCB		1362
	65		01	FB 00125	CALLS	#1, SYSSWAIT		
			5B	DD 00128	PUSHL	CCB		1363
	64		01	FB 0012A	CALLS	#1, SYSS\$DELETE		
			EA	11 0012D	BRB	20\$		1360
	05	08	AB	E8 0012F	BLBS	8(CCB), 22\$		1365
			37	DD 00133	PUSHL	#55		1367
	63		01	FB 00135	CALLS	#1, FOR\$\$\$SIGNAL_STO		
		00000000G	00	16 00138	JSB	FOR\$\$CB_POP		1374
			50	D4 0013E	CLRL	R0		1377
				04 00140	RET			
			0000	00141	.WORD	Save nothing		1208
	50	08	AC	D0 00143	MOVL	8(AP), R0		
	50	04	A0	D0 00147	MOVL	4(R0), R0		
		F8	A0	9F 0014B	PUSHAB	L_ERR_EQL PRES		
		FC	A0	9F 0014E	PUSHAB	L_UNWIND_ACTION		
			02	DD 00151	PUSHL	#2		
			5E	DD 00153	PUSHL	SP		
	7E	04	AC	7D 00155	MOVQ	4(AP), -(SP)		
00000000G	00		03	FB 00159	CALLS	#3, FOR\$\$\$IOSTAT_HND		
			04	00160	RET			

; Routine Size: 353 bytes, Routine Base: _FOR\$CODE + 00C4

FOR\$DELETE
1-002

K 16
16-Sep-1984 00:17:29
14-Sep-1984 12:31:48

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORDELETE.B32;1

Page 14
(5)

: 465 1378 1 END
: 466 1379 0 ELUDOM

!End of module

PSECT SUMMARY

:
: Name Bytes Attributes
: _FOR\$CODE 549 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

:
: File ----- Symbols ----- Pages Processing
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 Total Loaded Percent Mapped Time
: 9776 13 0 581 00:01.1

COMMAND QUALIFIERS

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:FORDELETE/OBJ=OBJ\$:FORDELETE MSRC\$:FORDELETE/UPDATE=(ENH\$:FORDELETE
:)

: Size: 549 code + 0 data bytes
: Run Time: 00:16.4
: Elapsed Time: 00:55.0
: Lines/CPU Min: 5045
: Lexemes/CPU-Min: 33318
: Memory Used: 185 pages
: Compilation Complete

0179 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COMR50WD
LIS

FORDATEDS
LIS

FORDECOMO
LIS

FORB
LIS

COMSETST
LIS

FORASSOC
LIS

FORCLOSEF
LIS

FORDATE
LIS

FORCLOSE
LIS

FORDECOMP
LIS

FORDELETE
LIS

COMRAD50
LIS

COMUSEREX
LIS

FORBITOPS
LIS

FORDEFINE
LIS

FORBACKSP
LIS

FORDISPA
LIS

FORCUTR
LIS